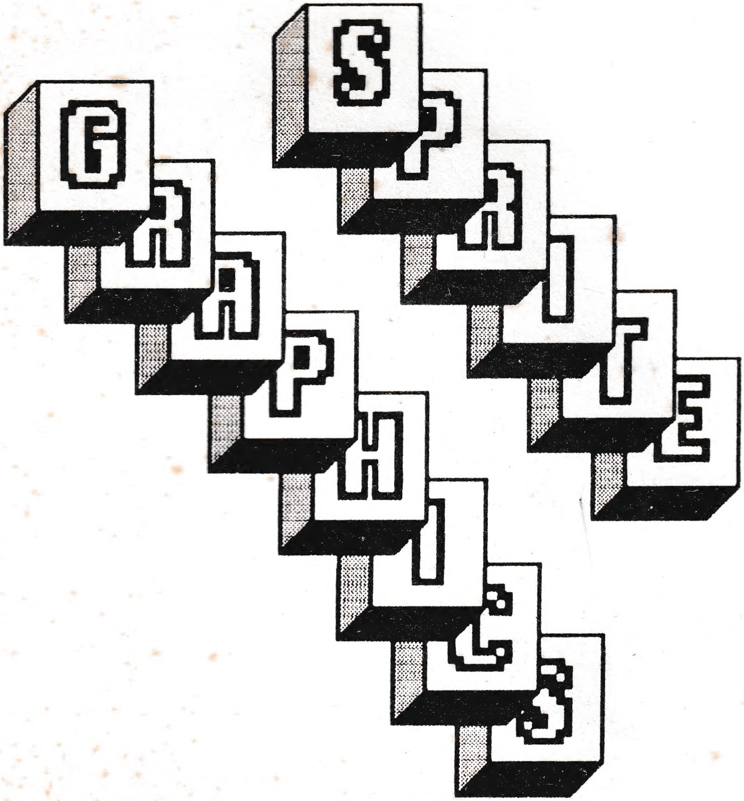
GRAPHICS
SPRITE

2R PROJECTS

AUSTRALIAN
VIDEO
PRESENTATIONS



AUSTRALIAN VIDEO PRESENTATIONS

Customer Satisfaction

Should you discover any innate defects in the documentation or software operation or hardware design within ninety (90) days from the date of purchase, Australian Video Presentations will replace or repair the faulty part at no charge providing the enclosed guarantee card has been filled in and returned to Australian Video Presentations within seven (7) days from the date of purchase.

Limitation on Warranties & Liability

Even though Australian Video Presentations has tested the software described in this manual and reviewed its contents and tested the associated hardware neither Australian Video Presentations nor Apple nor anyone else who has been involved in the creation, production or delivery of the Sprite Graphics system make any warranty or representation, either express or implied, with respect to this manual or the software described in this manual or the associated hardware, their quality, performance, or fitness for any particular purpose, nor shall they be liable for any direct, incidental or consequential damages, loss of anticipated profits or benefits resulting from any defect in the software or manual or hardware. This Sprite Graphics system is sold "as is", and you the purchaser are assuming the entire risk as to its performance.

Copyright

At the time of going to press this manual and software described in this manual and that software named Spritesoft and Paintsoft have copyright applications pending. The Sprite Graphics hardware has Patent applications pending. This manual and the programs must not be copied, in whole or part, without the written consent of Australian Video Presentations, except in the normal use of software or to make a backup copy. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased may be sold or loaned to another person. You may use this software and hardware on any computer owned by you but extra copies cannot be made for this purpose.

Limited Warranty for Diskettes

Australian Video Presentations warrants the diskette or diskettes on which the software is recorded to be free from defects in materials and faulty workmanship under normal use and storage for a period of thirty (30) days after the date of purchase. If during this period a defect in the diskette or diskettes should occur, the diskette or diskettes may be returned to Australian Video Presentations or an official dealer for replacement, provided you have completed and returned to Australian Video Presentations the warranty card within seven (7) days from the date of purchase.

Hardware Warranty

Australian Video Presentation warrants the Sprite Graphics hardware to be free from defects in materials and faulty workmanship under normal usage and storage for a period of twelve (12) months from the date of purchase provided that all installation and operating procedures as described within this manual are adhered to. If during this period a defect should occur, the hardware may be returned to Australian Video Presentations or an official dealer for repair, provided that you have completed and returned to Australian Video Presentations the warranty card within seven (7) days from the date of purchase.

Trademarks

"APPLE" & "Applesoft are trademarks of Apple Computer Incorporated.

ACKNOWLEDGMENTS

The management of Australian Video Presentations and 2R Projects Pty. Ltd. would like to thank the following individuals who together formed the Sprite Graphics design team:

Wolf Steinke
Noel Coward
Rick Spielrein
Mike Percy
Martin Thompson

and gratefully acknowledges the assistance of Metropolitan Business Machines, John Stachlewski, Darryl Annett, Jamie Walls and James Guest.

INTRODUCTION

WELCOME TO SPRITE GRAPHICS

Sprite Graphics is an animation and graphics package for the Apple and is unsurpassed in its ease of operation and colour display. With Sprite graphics you can create your own games or graphic displays, experiment with vectors and angles; incorporate animation into your programmes and paint pictures on the screen.

Spritesoft is a language written especially for Sprite graphics. It helps you to create Sprites and then animate them. It can make your Sprites walk, fly or run around the screen. All of this can be used within Applesoft programmes.

Paintsoft is a painting programme written in Spritesoft and Applesoft. It enables you to paint a picture on the screen with the aid of a joystick. You can paint in sixteen colours with a normal paint brush or an air brush or any other type of brush that you wish to create.

This manual has been written only as an aid to learning Sprite graphics. By this we mean that you should sit down with both the computer and this manual and use both together. Make sure you read the instructions on the screen (prompts). If these are followed you may not need to read certain sections of this manual. The best way to learn Sprite graphics is by using the system, so go ahead and enjoy it.

CONTENTS

SPRITE GRAPHICS PACKING LIST

Your PAL or NTSC system should contain:

1 Sprite Graphics card.
1 Spritesoft Systems disk.
1 Sprite Graphics manual.
1 Warranty card.
1 Video lead (NTSC SYSTEM ONLY).

Your RGB system should contain:

1 Sprite Graphics card.
1 6 pin to 6 pin video cable.
1 Spritesoft Systems disk.
1 Sprite Graphics manual.
1 Warranty card.

NOTE : The warranty card must be filled in and posted to Australian Video Presentations within 7 days from the date of purchase or the warranty will become void.

Hardware Requirements.

To use this Sprite system you need :

An AppleII, II+, IIe or Europlus with 48K RAM.
One disk drive and controller.
One monitor.
One joy stick .

INSTALLATION

CAUTION! TURN THE COMPUTER OFF!

Spritesoft assumes that you have at least one disk drive plugged into slot six of the computer. The Sprite card itself is **slot independent**, that means that it can be placed into any available slot.

IMPORTANT!

ALWAYS BE SURE THAT THE SPRITE GRAPHICS CARD IS IN A SLOT BEFORE CONNECTING OR DISCONNECTING ANY CABLES TO IT!

RGB SYSTEM OWNERS ONLY:

Your RGB card has an on board softswitch. This means that when you are using the Apple's video output that is what will appear on the screen. However when you start to use the video output of the Sprite card the softswitch will automatically switch the screen to the output of the Sprite card.

By now you are probably wondering what the switch on the end of the cable is for. This is used manually to switch the monitor's screen from the Sprite output to the computers output and vice versa (from the computer to the Sprite output).

The video output of the Sprite RGB card is a standard 6 pin connector used by most manufacturers of RGB cards for the Apple. Take the video lead that is usually plugged into the output of your RGB card and plug it into the connector on the Sprite Card marked RGB OUTPUT. Now take the RGB cable supplied with the Sprite card and plug one end into the connector on the Sprite card marked RGB INPUT . Then plug the other end into you RGB card's output connector. If nothing appears on the screen when you turn the computer on try switching the switch on the end of the cable. If there is still nothing on the screen you have probably plugged the six way cable in back to front. Try reversing one end and you should now see something on the screen.

PAL SYSTEM OWNERS ONLY:

If you are fortunate enough to own two monitors then you can connect your Sprite card to one monitor and the output of your Apple to the other. If you have only one monitor, plug the output of the Sprite card into the monitor when running a Sprite graphics programme and the Apple video output in when writing or editing Sprites and their programmes.

A video softswitch made by the manufacturer of the Sprite card is available from the dealer who sold you the Sprite card. This automatically switches the monitor to either the Apple output, the output of the Sprite card or the output of another peripheral card eg. an 80 column card. As you may discover, the softswitch is invaluable when working with only one monitor.

Your NTSC card has an on-board softswitch. This means that when you are using the Apple's video output that is what will appear on the screen. However when you start to use the video output of the Sprite card the softswitch will automatically switch the screen to the output of the Sprite card.

By now you are probably wondering what the switch on the end of the cable is for. This is used manually to switch the monitor's screen from the Sprite output to the computers output and vice versa (from the computer to the Sprite output).

Supplied with your Sprite card is a video input lead. The end with the flat connector plugs into the connector on the Sprite card marked INPUT. The other end, with the round RCA connector, plugs into the video output of the Apple. The lead from your monitor which is usually connected to the computer plugs into the connector on the Sprite card marked VIDEO OUTPUT.

CONFIGURING THE SOFTWARE.

If you have not yet installed the Sprite card into the computer you should do so now. If you do not know how to do this you should read the page before this.

Having successfully installed the hardware you are now ready to install the software.

1. Place the Spritesoft disk in drive 1 and close the door.
2. Turn the monitor or TV on. (Wait for it to warm up.)
3. Turn the computer on.

Watch and Listen.

The red light on the disk drive should have come on and the disk drive made a whirring sound for about 15 seconds. Next, the Australian Video Presentations logo should have been displayed in normal then in inverse. You should then have been left with your first selection :

```
<1>  SINGLE  DISK  DRIVE
     <2>  DUAL  DISK  DRIVE

     SELECT  < >
```

Reply by selecting the appropriate **prompt** (Number). Eg. 1 or 2 .

> (A **Prompt** is either a number or letter preceding each possible choice or alternative offered to you. In the example above the numbers <1> and <2> are used to prompt you to choose one of the alternatives by selecting the number 1 or the number 2.)

After making your selection the red light should have come on again and the drive started whirring until the main menu appeared on the screen. The main menu should look the same as the diagram on page 3.0 .

Spritesoft must know into which slot you have installed the hardware in order for it to access the hardware correctly. To **configure** Spritesoft select the prompt **<P>** for PATCH SPRITE ROUTINES from the main menu.

> (To **Configure** the software means to organise the software so that it can interact correctly with the hardware.)

After selecting <P> the disk drive once again whirrs away and a new set of prompts appears :

<1> PATCH SPRITE ROUTINES
<2> PATCH SPRITESOFT

SELECT < >

Select prompt <1>. Another prompt appears on the screen :

SLOT NUMBER OF SPRITE CARD. < >

Type in the number of the slot into which the Sprite card has been inserted. If you are not sure of the slot number ,you will find that the number is printed on the Apple's main board at the top of each slot.

Having done this you will see the computer printing the file conversion on the screen as it carries out (executes) each step. When completed the main menu will appear as the diagram on page 3.0 . Once again select the PATCH routine and repeat the above procedure, this time selecting **<2> PATCH SPRITESOFT**. When completed, the main menu will once again appear on the screen. You have now finished configuring Spritesoft and you are ready to continue to the next section which will show you how to display the entire colour set on the Sprite's video output.

For those more technically minded:
The Patch routine is a machine code program designed to patch those absolute addresses in the Sprite routines which are used to address the Video Display Processor (VDP) directly. It then unlocks, BSAVES and relocks all the relocatable (.REL) type files on the disk.

ADJUSTING THE COLOUR

Having successfully installed the Sprite card you should now check its colour. In order to do this the main menu must be displayed on the screen as shown on page 3.0 . If this is not displayed select from the menu the option <Q> QUIT . If this fails turn the computer off, wait two seconds and turn the computer on again, enter the number of disk drives as shown previously and the next display will be the main menu.

Each Sprite card has been aligned on a computer at the factory to produce correct colour. Unfortunately all computers are not identical. This means that when used on your computer, the Sprite card may need adjusting to enable it to produce the correct colours. A set-up routine containing a test pattern has been included on the disk to help you test and if necessary adjust the colours.To run this set-up routine select:

<S> SET-UP

then

<2> SPRITE CARD

You are now asked or prompted by the computer to select either a colour bar display, a pallette type display or quit this routine and return to the main menu. The colour bar display suites our purpose so select :

<C> COLOR BARS

The output of the Sprite card will now be displaying a colour bar test pattern. Remember that if you are using an RGB or NTSC system you must use the switch on the end of the lead to change the monitor display to and from the output of the Sprite card. Alternatively if you are using a PAL system you may need to unplug the monitor from the computer and plug it into the Sprite card; this is provided you do not have a soft switch connected to the PAL Sprite card.

The following two pages explain how to align each Sprite card depending on the type of system , ie. PAL or RGB. NTSC owners do not have the ability to adjust the colour as the Sprite graphics processor does this internally.

RGB SYSTEM OWNERS

You will see on your RGB card that there are three variable resistors or trimpots named VR1, VR2 and VR3. They are near the bottom edge of the Sprite card near the connector which plugs into the computer. These can be adjusted by rotating them with a small screwdriver. Each trimpot is labelled with the name of the colour that it controls, eg. VR1 controls the amount of red in the picture. By turning the trimpot clockwise (looking from the connector side) you will **decrease** the amount of red in the picture.

To adjust the colour simply look at the colour bar test pattern and decide if there is too much of one colour or not enough of another. **In most instances no adjustment should be necessary.** If an adjustment is needed, turn off the computer and disconnect the monitor. Then carefully unplug the card and turn the appropriate trimpot in the appropriate direction. Plug the card back into the computer and display the colour bars again. If further adjustment is required repeat the above procedure until the colours are correct.

Either the colour bar test pattern or the pallette of colours may be used for the above procedure. We suggest that you display the colour bars first then refer to the colour pallette when you think the colours are correct.

PAL SYSTEM OWNERS

You will see on your PAL card that there are four variable resistors or trimpots; three are grouped together and one is by itself near the connector that plugs into the computer. To help identify them they are labelled VR1, VR2, VR3 and VR4 respectively. These trimpots are used to adjust the balance of colours and the syncronization of the video signal. Adjustments are made by turning the trimpots with a small screwdriver.

If the display is not in colour trimpot number 2 (VR2) should be adjusted first. Mark its original position with a felt pen. Now, with the computer turned on, carefully turn VR2 slightly clockwise. If no colour appears return it to its original position and then turn it in the opposite direction by the same amount. If this has no effect return it to the original position and repeat this procedure with VR1.

If the display is still not in colour repeat the above procedure with VR4 after returning VR2 and VR1 to their original positions.

By now you will have a colour display but the colour balance may not be correct, eg. the test pattern may look too red. To adjust for a correct colour balance turn VR1 fully clockwise and then turn it back towards its previous position, at the same time looking at the colour bars displayed. Stop turning VR1 when the white section of the colour bars look most white, eg. when the red or blue or green tinge reduces to a minimum

Now repeat the colour balance adjustment using VR2. However this time turn VR2 fully anti-clockwise and then work back towards its previous position.

Either the colour bar test pattern or the pallette of colours may be used for the above procedure. We suggest that you display the colour bars first and then refer to the colour pallette when you think the colours are correct.

SPRITES, WHAT ARE THEY?

If you are not already familiar with Sprites you should read this section before continuing.

Sprites can best be imagined as a set of planes or layers sandwiched together. There are thirty two Sprite planes in all, 0 to 31. Turn to the diagram on page 2.2 and you will see a graphic example of Sprite planes.

Back again?

As you have seen, each plane can hold **one Sprite in one colour only**. The Sprites on the planes closest to the viewer, plane 0 being the closest, have higher priority than those further away. This means that **a Sprite on plane one will superimpose (mask) over any other Sprites in the same position on a plane with lower priority**. If you find this a bit hard to understand, turn to pages 2.3 to 2.5 and you will find a graphic example of this.

Back again?

Did you notice how useful this automatic priority structure is for making cars look like they are driving in front of hills and behind trees. This graphic scene used two Sprites, one Sprite for the tree on plane one and one Sprite for the car on plane two.

You have seen Sprites but we have not yet defined them. **Sprites** are made up of graphic blocks and **are available in three sizes : 8 x 8 blocks, 16 x 16 blocks or 32 x 32 blocks**. On page 3.6 you will find a grid used for designing 8 x 8 Sprites. On page 12.0 is another design grid on which you may wish to design your Sprites with the aid of a water base pen.

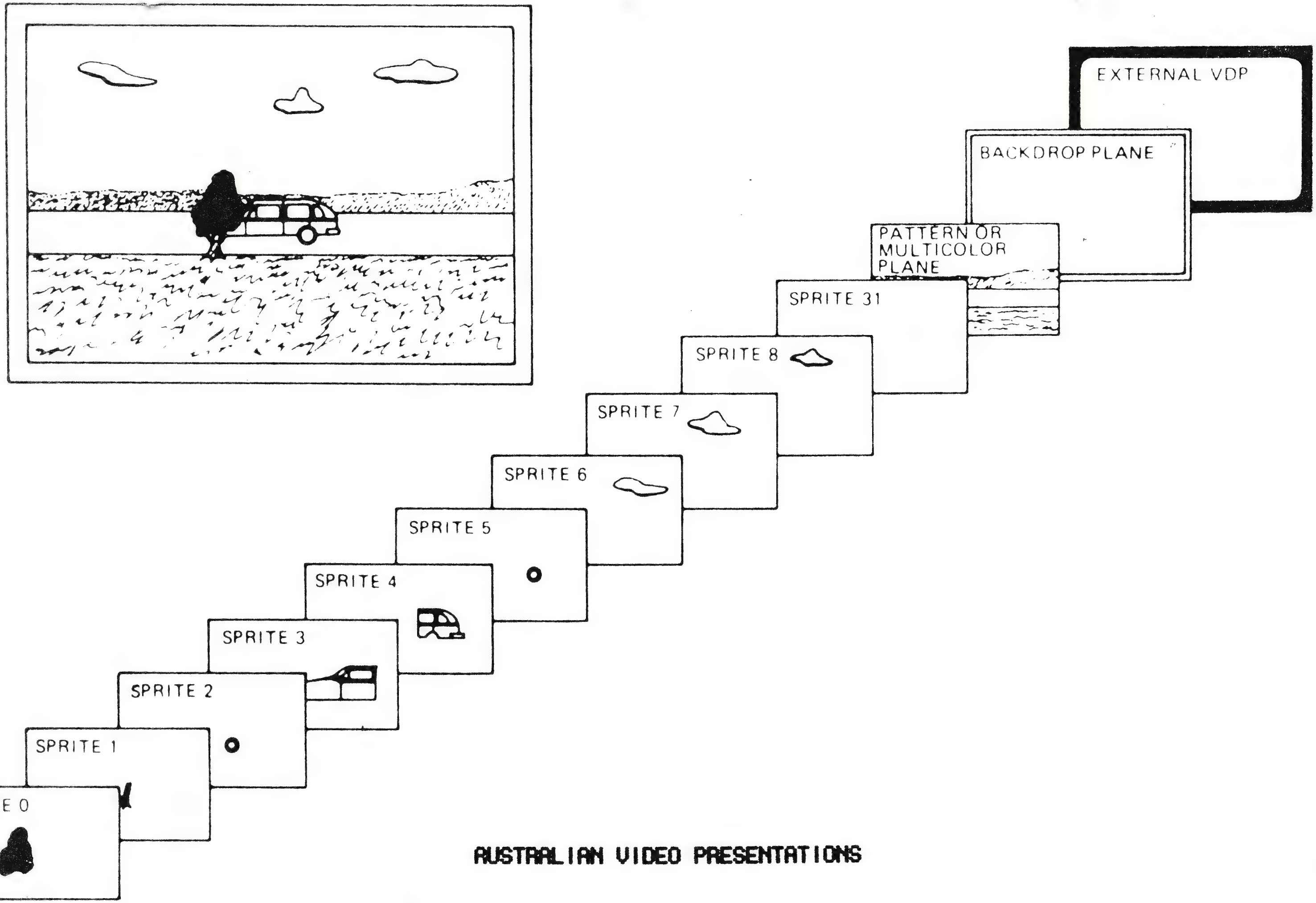
So far you have learned that Sprites are shapes made up on any one of three grid sizes. These shapes do not become Sprites until you have given them (defined) a colour and a position on the screen (co-ordinates). **The Sprites' colour and position are called its Attributes.** To avoid confusion we shall refer to Sprites with no attributes as Patterns, not Sprites.

It is important to remember the difference between Sprites and Patterns as some programmes described in this manual will work only with Patterns. Other programmes will work with both Patterns and Sprites.

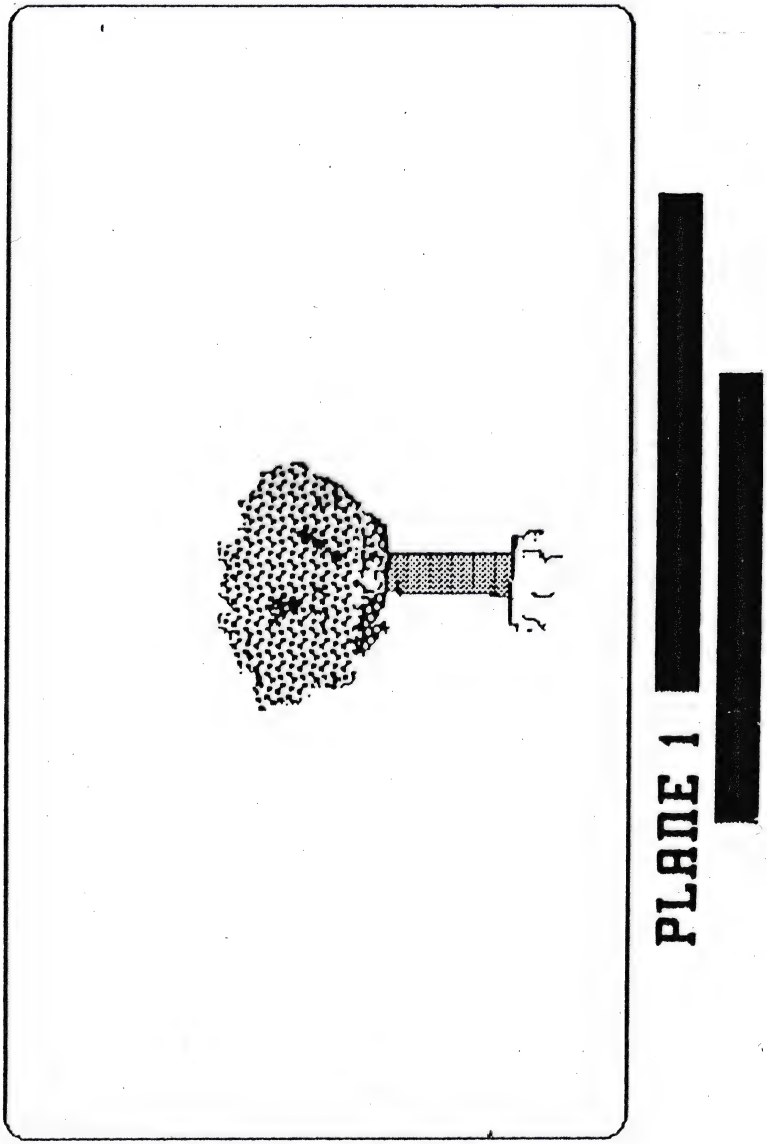
Behind the Sprite planes is the Pattern plane, see page 2.5 . This, as suggested by its name is used for drawing patterns. The screen format or size of the Pattern plane is 256 x 192 pixels or dots. Horizontal positions or co-ordinates are defined as a number from 0 to 255; the vertical co-ordinates being a number from 0 to 191. The capabilities and method of using this plane are fully explained in section 8 of this manual.

The Backdrop plane follows the Pattern plane, see page 2.2 . It is larger than the previous planes and we use the **Backdrop plane** to **colour in the area behind the Sprites.** Any of **sixteen colours** are available to colour either the Backdrop plane or the Sprites. For a list of these colours turn to page 9.7 .

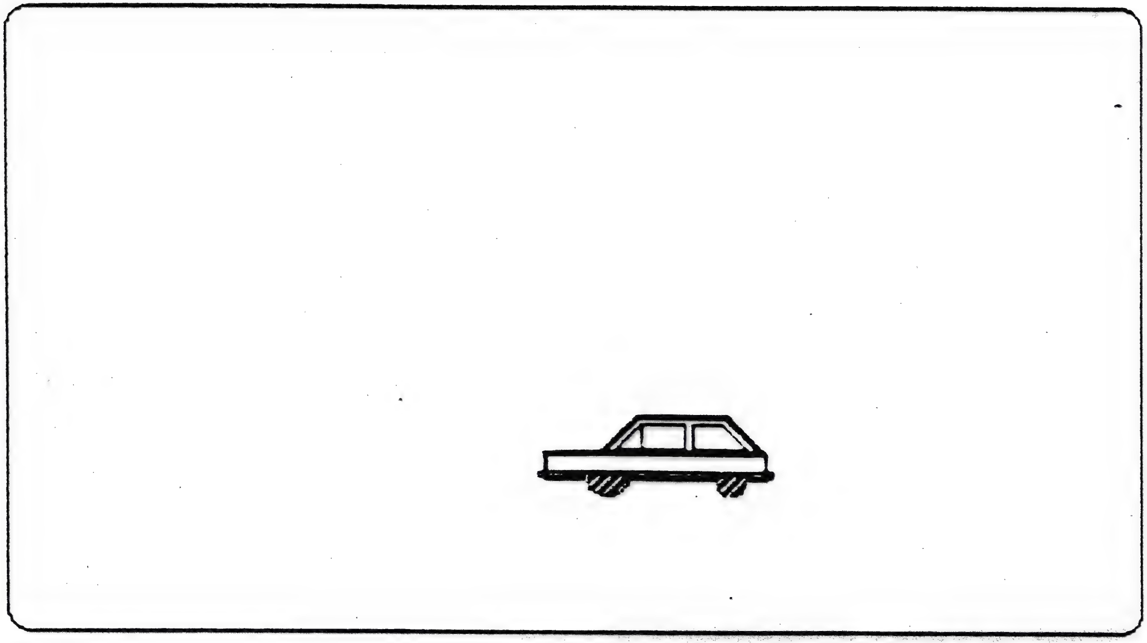
SECTION 2.2



AUSTRALIAN VIDEO PRESENTATIONS

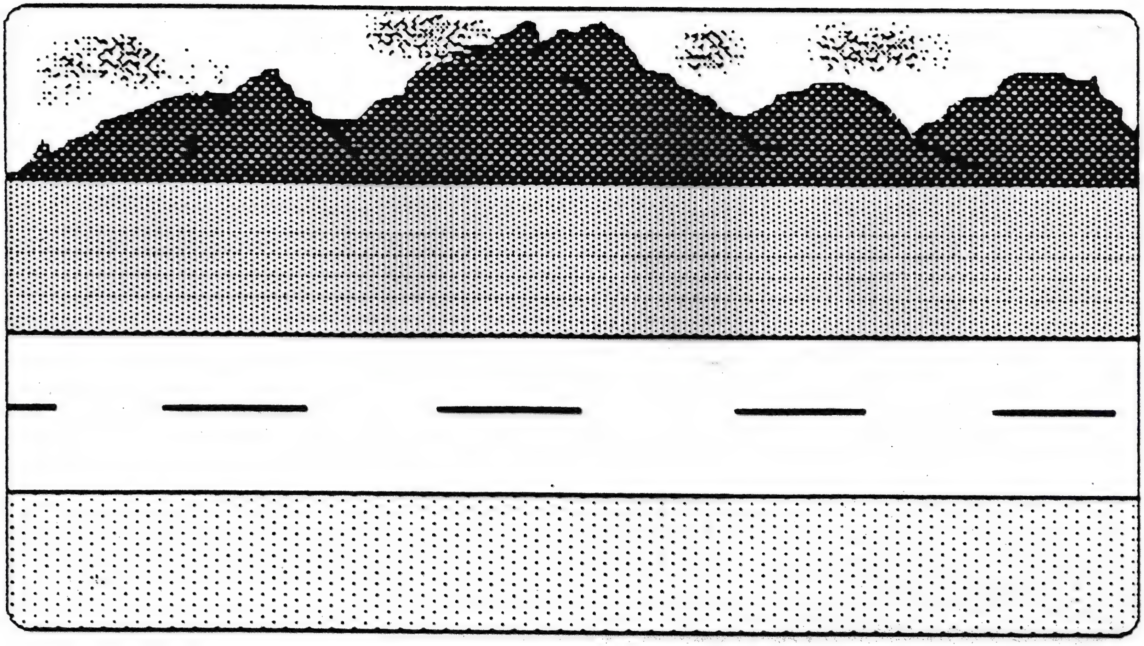


SECTION 2.3



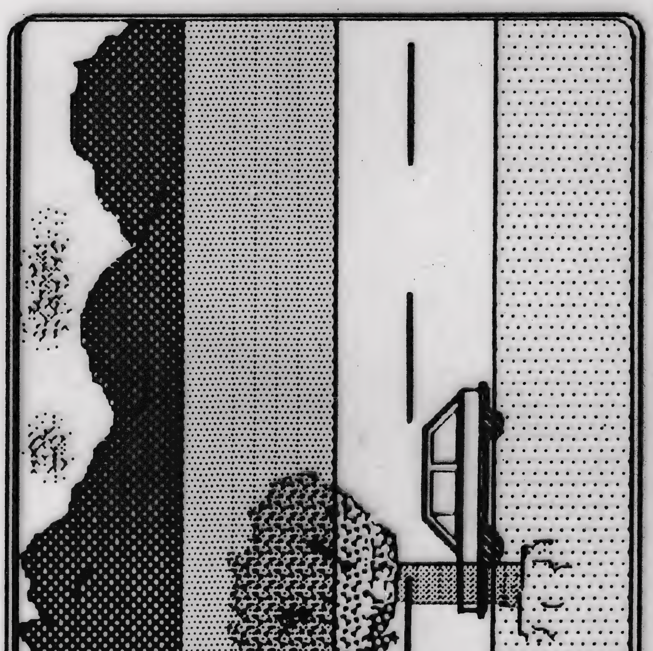
PLANE 2

SECTION 2.4



PATTERN PLANE

SECTION 2.5

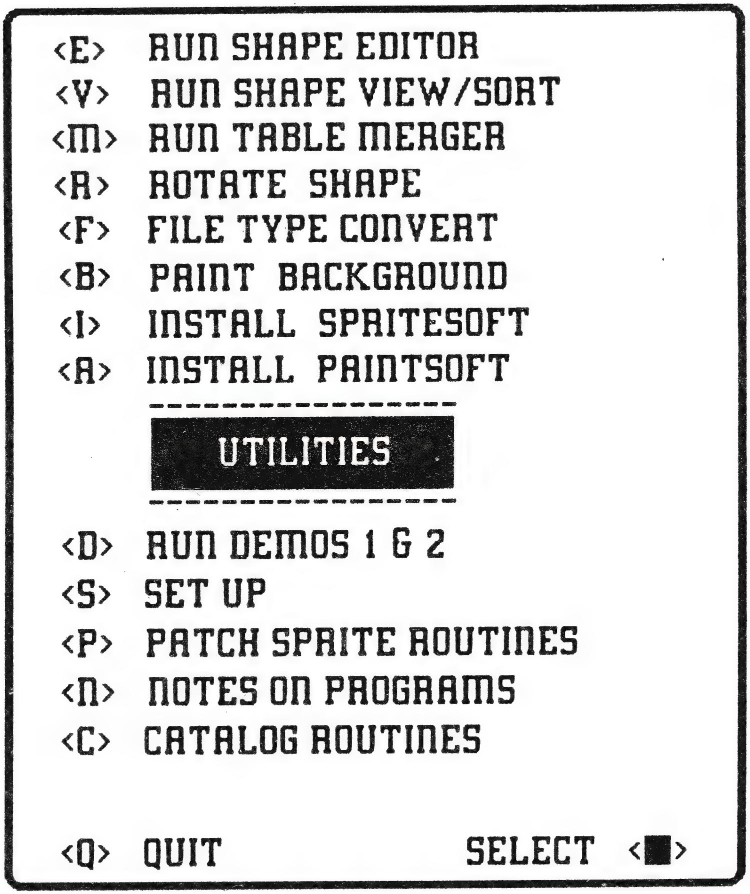


PLANE 1

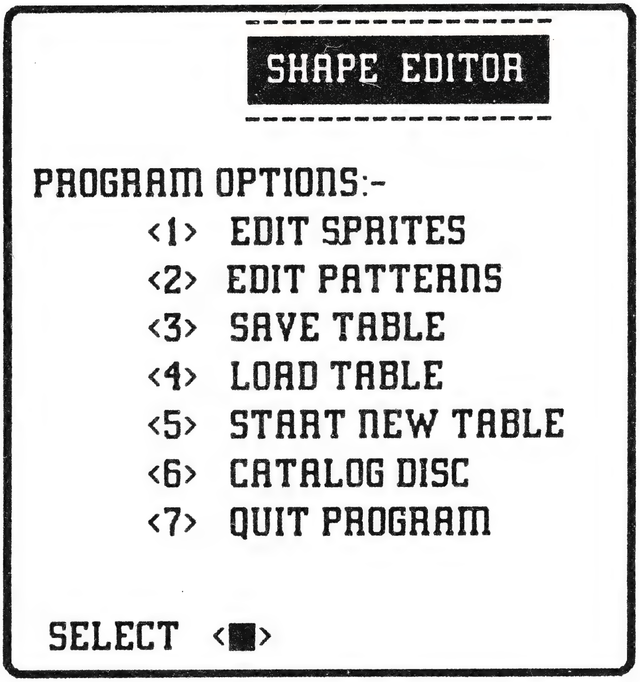
[redacted]

[redacted]

SECTION 2.3 SECTION 2.4 SECTION 2.5



```
<E>   RUN SHAPE EDITOR
<V>   RUN SHAPE VIEW/SORT
<M>   RUN TABLE MERGER
<R>   ROTATE  SHAPE
<F>   FILE TYPE CONVERT
<B>   PAINT  BACKGROUND
<I>   INSTALL  SPRITESOFT
<A>   INSTALL  PAINTSOFT
      -----------------
           UTILITIES
      -----------------
<D>   RUN DEMOS 1 & 2
<S>   SET UP
<P>   PATCH SPRITE ROUTINES
<N>   NOTES ON PROGRAMS
<C>   CATALOG ROUTINES


<Q>   QUIT                    SELECT  <█>
```



```
SHAPE EDITOR

PROGRAM OPTIONS:-
        <1>  EDIT SPRITES
        <2>  EDIT PATTERNS
        <3>  SAVE TABLE
        <4>  LOAD TABLE
        <5>  START NEW TABLE
        <6>  CATALOG DISC
        <7>  QUIT PROGRAM

SELECT  <■>
```

In order to be able to create shapes you must have already installed the hardware and configured the software for your system. If you have not done this or you do not know what this means you should read the previous pages on installation and software configuration before proceeding.

If you have successfully carried out the above operations then you are ready to create and edit shapes. With the aid of the Shape Editor routine shapes are very easy to create. The Shape Editor is the first routine listed on the main menu. If the main menu is not displayed on the screen either turn your system on if it is off (this is called a cold boot.), or select from the menu on the screen the option which says "QUIT PROGRAM". This will take you out of the programme that you are presently in and return you to the main menu. Now select :

<E> RUN SHAPE EDITOR

After a few moments the Shape Editor menu will appear. See the diagram on page 3.1 . Choosing option <1> or <2> will select the type of shape that you will be editing and take you into the create mode. Flow Chart number 1 on page 3.5 shows all the possible sequences that you may follow while in this program.

Plotting on the grid is achieved by using the cursor to draw the desired shape. Letters **A** and **Z** move the cursor up and down respectively while the <- and -> arrows move the cursor left and right respectively. The space bar switches the cursor between plotting and moving without plotting. Finally letter **X** or **CTRL X** erases plotted points. These are the only six actions that you need to remember to be able to create and edit shapes.

The diagram on page 3.6 summarises these codes and can readily be referred to when in doubt. You may find it easier to plan your shapes on a worksheet of graph paper first. On page 12.0 you will find a worksheet with grids printed onto a plastic page. You can draw your shapes with a water base pen and erase them when you have finished.

Once you have created the shape and entered it into the computer you are asked by the computer (prompted) if you wish to:

ADD ATTRIBUTES NOW?

If you select **NO** you must define the patterns' colour and co-ordinates when using it in Spritesoft. If you don't you will not be able to see the pattern and you will not know where it is on the screen.

If you select **YES** you must next define its attributes. Colour is the first attribute to be added. A list of colours and their defined numbers can be found on the top of page 3.5 or on page 9.7 . Next to be added is the vertical and horizontal co-ordinates. These are defined on page 2.1 . If you are finding attributes a little hard to understand try reading pages 2.0 and 2.1 again.

The best way to learn is by doing! Try creating some shapes, save them to disk and then reload them. Try the other options in the Shape Editor menu. If you are not sure what the screen display is indicating then refer to the flow chart to see where you are and what to do next. The worst that can happen is that you may lose the shapes that you have created.

After creating, saving or loading shapes you may have noticed that the top line of text (just below the grid) changes. The number displayed in inverse in the centre of the screen tells you how many shapes are in the file that you have loaded or created. The right hand number indicates the shape number that is presently being displayed or that will be decoded and displayed if return is selected. When creating shapes you must increment (see paragraph below) this number each time you enter a new shape. If you don't the new shape will erase the previous one.

The <- and -> arrow keys can be used to increment or decrement (count up or down) to the shape number that is to be decoded from the loaded file. When working with very large files the desired shape number to be decoded may be entered directly from the keyboard.

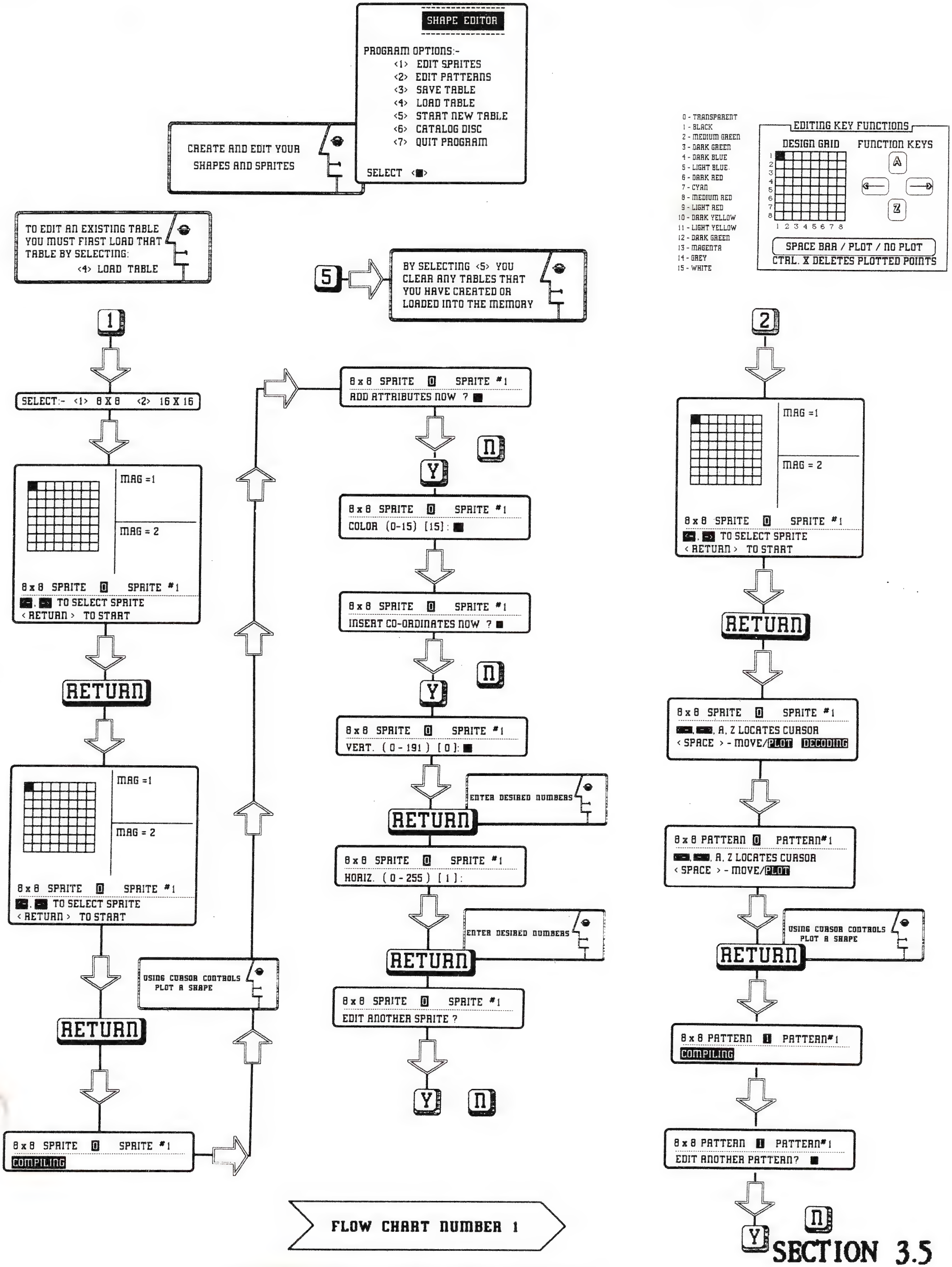
TRY THIS!

Try this example of Spritesoft. You may not yet understand what it means, however it will help you in understanding the next section of the manual.

From the main menu select <I> to Install Spritesoft then type in the following:

&C,5 (Return)	Colours the background to colour number 5, light blue.
&L,"D2B.SPR" (Return)	Loads the file named D2B.SPR into memory.
&T,16 (Return)	Tells the computer that you are using a 16x16 Type Sprite.
&S,1,1 (Return)	Sets Sprite number 1 on plane number 1.
&EL (Return)	Enlarges the 16x16 Sprite to become a 32x32 Sprite.
&S,2,2 (Return)	Sets Sprite 2 on plane 2.
&S,6,3 (Return)	Sets Sprite 6 on plane 3.
&S,7,4 (Return)	Sets Sprite 7 on plane 4.
10 &ML,3,4,1 (Return)	This is the first line of an Applesoft programme that moves Sprites 3 & 4 to the left by one step.
20 GOTO 10 (Return)	This second line of the programme repeats the action of the first line.
RUN (Return)	Tells the computer to run the programme.

You should now see a car drive across the screen from right to left. Remember that if you have an RGB or NTSC system the Sprite screen will be switched to automatically when you run the programme. PAL users will need to use two monitors or a soft switch.



```
SHAPE EDITOR

PROGRAM OPTIONS:-
        <1>  EDIT SPRITES
        <2>  EDIT PATTERNS
        <3>  SAVE TABLE
        <4>  LOAD TABLE
        <5>  START NEW TABLE
        <6>  CATALOG DISC
        <7>  QUIT PROGRAM

 SELECT  <■>
```

CREATE AND EDIT YOUR SHAPES AND SPRITES

TO EDIT AN EXISTING TABLE YOU MUST FIRST LOAD THAT TABLE BY SELECTING: <4> LOAD TABLE

5 ⇨ BY SELECTING <5> YOU CLEAR ANY TABLES THAT YOU HAVE CREATED OR LOADED INTO THE MEMORY

0 -	TRANSPARENT
1 -	BLACK
2 -	MEDIUM GREEN
3 -	DARK GREEN
4 -	DARK BLUE
5 -	LIGHT BLUE.
6 -	DARK RED
7 -	CYAN
8 -	MEDIUM RED
9 -	LIGHT RED
10 -	DARK YELLOW
11 -	LIGHT YELLOW
12 -	DARK GREEN
13 -	MAGENTA
14 -	GREY
15 -	WHITE

EDITING KEY FUNCTIONS

DESIGN GRID

FUNCTION KEYS

SPACE BAR / PLOT / NO PLOT

CTRL. X DELETES PLOTTED POINTS

1

```
SELECT:-  <1>  8 X 8    <2>  16 X 16
```

```
                  MAG =1

                  MAG = 2

8 x 8  SPRITE   0     SPRITE #1
▬, ▬  TO SELECT SPRITE
< RETURN >   TO START
```

RETURN

```
                  MAG =1

                  MAG = 2

8 x 8  SPRITE   0     SPRITE #1
▬, ▬  TO SELECT SPRITE
< RETURN >   TO START
```

USING CURSOR CONTROLS PLOT A SHAPE

RETURN

```
8 x 8  SPRITE   0     SPRITE #1
COMPILING
```

```
8 x 8  SPRITE   0     SPRITE #1
ADD ATTRIBUTES NOW  ? ■
```

N

Y

```
8 x 8  SPRITE   0     SPRITE #1
COLOR  (0-15)  [15]: ■
```

```
8 x 8  SPRITE   0     SPRITE #1
INSERT CO-ORDINATES NOW  ? ■
```

N

Y

```
8 x 8  SPRITE   0     SPRITE #1
VERT.  ( 0 - 191 )  [ 0 ]: ■
```

ENTER DESIRED NUMBERS

RETURN

```
8 x 8  SPRITE   0     SPRITE #1
HORIZ.  ( 0 - 255 )  [ 1 ]:
```

ENTER DESIRED NUMBERS

RETURN

```
8 x 8  SPRITE   0     SPRITE #1
EDIT ANOTHER SPRITE ?
```

Y N

2

```
                  MAG =1

                  MAG = 2

8 x 8  SPRITE   0     SPRITE #1
▬, ▬  TO SELECT SPRITE
< RETURN >   TO START
```

RETURN

```
8 x 8  SPRITE   0     SPRITE #1
▬, ▬, A, Z LOCATES CURSOR
< SPACE > - MOVE/PLOT  DECODING
```

```
8 x 8 PATTERN  0    PATTERN#1
▬, ▬, A, Z LOCATES CURSOR
< SPACE > - MOVE/PLOT
```

USING CURSOR CONTROLS PLOT A SHAPE

RETURN

```
8 x 8 PATTERN   1   PATTERN#1
COMPILING
```

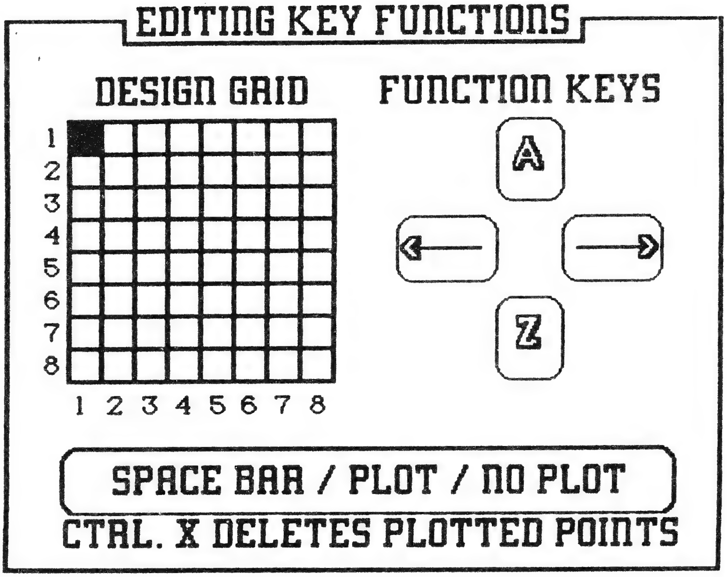
```
8 x 8 PATTERN   1   PATTERN#1
EDIT ANOTHER PATTERN?  ■
```

Y N

FLOW CHART NUMBER 1

SECTION 3.5

AUSTRALIAN VIDEO PRESENTATIONS



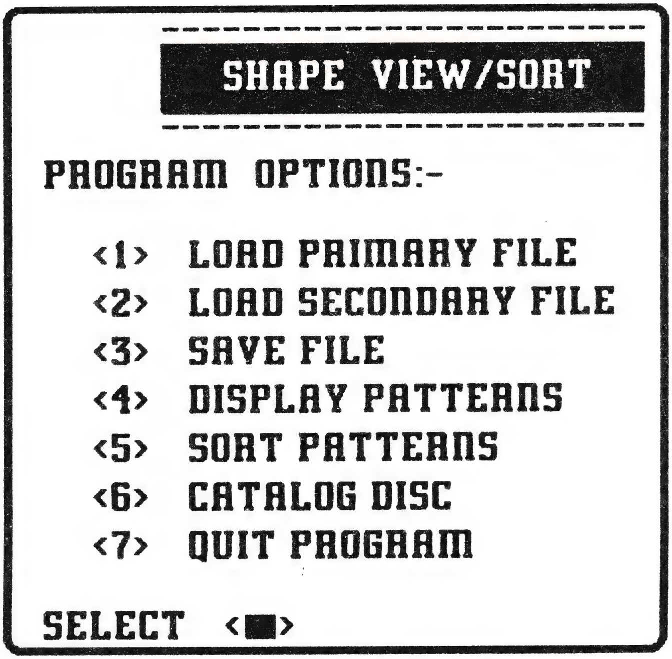
EDITING KEY FUNCTIONS

DESIGN GRID

FUNCTION KEYS

SPACE BAR / PLOT / NO PLOT

CTRL. X DELETES PLOTTED POINTS



```
        SHAPE VIEW/SORT

PROGRAM OPTIONS:-

   <1>  LOAD PRIMARY FILE
   <2>  LOAD SECONDARY FILE
   <3>  SAVE FILE
   <4>  DISPLAY PATTERNS
   <5>  SORT PATTERNS
   <6>  CATALOG DISC
   <7>  QUIT PROGRAM

SELECT  <■>
```

SHAPE VIEW SORT

The Shape View/Sort programme is menu driven. This means that you are prompted by the screen display, just like the Shape Editor which you have been using. In this mode there are two working files that may be loaded and accessed. These are the **Primary** and **Secondary** files

The two main utilities of this program are :

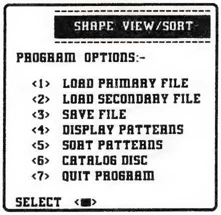
<4> DISPLAY PATTERNS
<5> SORT PATTERNS

As the name suggests, number 4 is used to display the patterns that you have created, or those that come with the software. (Any file listed in the catalogue that ends in: .SPR or .PAT) Also available in this programme is a print routine. By selecting <P> the page of Sprites displayed on the screen will be dumped to your printer and printed out. (This assumes that you have a standard Epson dot matrix printer connected to slot one.)

Number 5 is used to sort the patterns into any desired order. It may also be used to duplicate patterns within a table. When sorting the patterns you will find a U shaped cursor appears. With the aid of the cursor control this cursor is placed on top of the shapes that are to be deleted or swapped. The cursor is also used to mark the position at which point the duplication is to start and finish.

Once again, the best way to learn is by doing. Try out each of the utilities in turn. If you are not sure what the screen display is telling you refer to Flow Chart number 2 on page 4.2 . There are a couple of points that you should be aware of to avoid frustration and confusion :

- * You must first load a file before using routines 4 or 5.
- * Both save and sort routines can only be used with the Primary file
- * The space bar is used to toggle (switch) between displayed video pages. This is used when tables exceed 128 shapes for 8x8 shapes and 60 shapes for 16x16. (Maximum of 64 Sprites between both pages in 16 x 16 mode.)
- * The ESC key may be used (if no other key is specified) to return to the sub menu. (This is the Shape View/Sort menu)

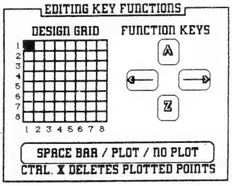


SHAPE VIEW/SORT

PROGRAM OPTIONS:-

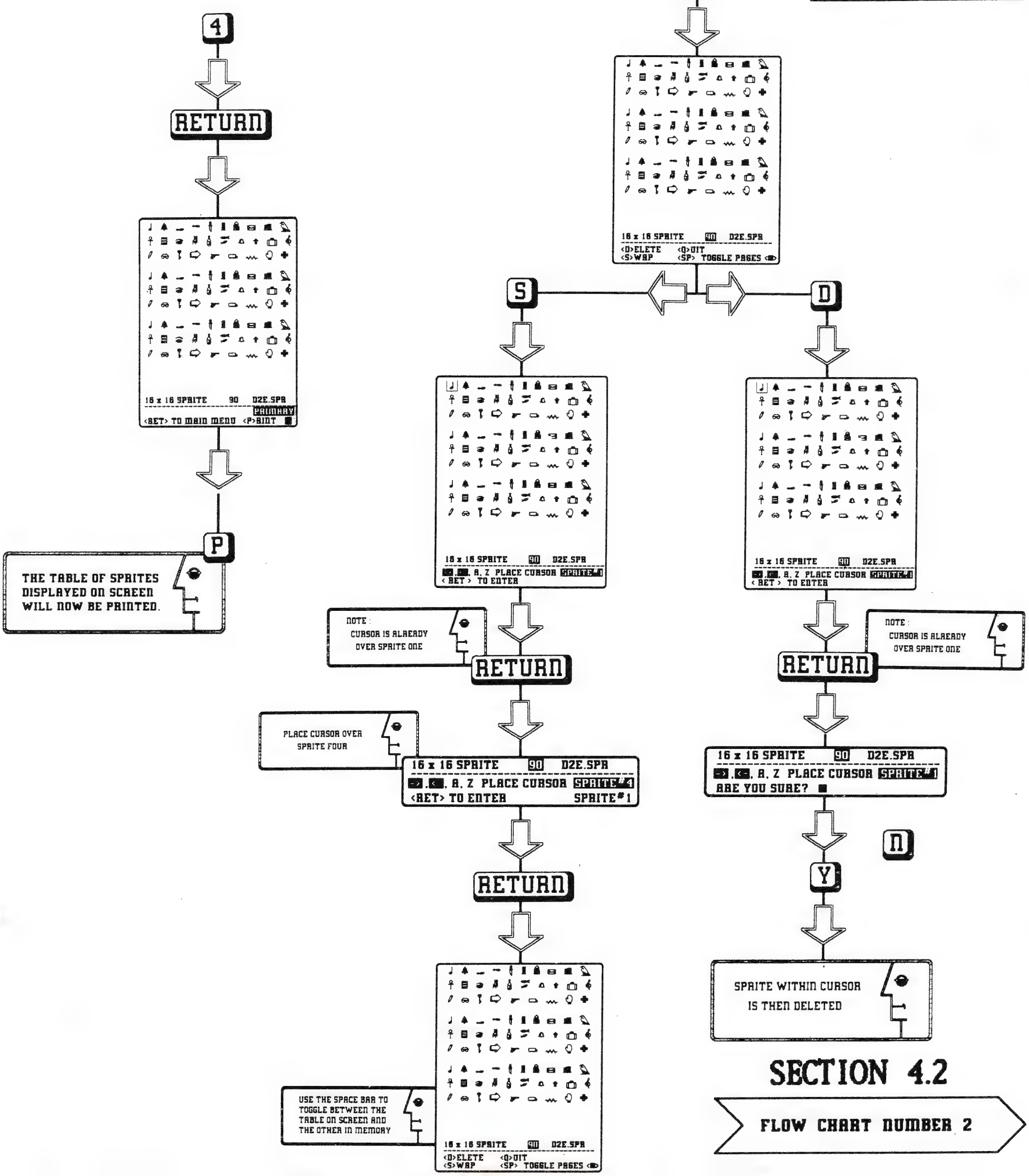
```
<1>  LOAD PRIMARY FILE
<2>  LOAD SECONDARY FILE
<3>  SAVE FILE
<4>  DISPLAY PATTERNS
<5>  SORT PATTERNS
<6>  CATALOG DISC
<7>  QUIT PROGRAM

SELECT  <■>
```

LOAD EITHER PRIMARY OR SECONDARY FILES



EDITING KEY FUNCTIONS

SPACE BAR / PLOT / NO PLOT
CTRL. X DELETES PLOTTED POINTS



```
16 x 16 SPRITE      90     D2E.SPR
                          PRIMARY
<RET> TO MAIN MENU   <P>RINT  ■
```

THE TABLE OF SPRITES DISPLAYED ON SCREEN WILL NOW BE PRINTED.

```
16 x 16 SPRITE      90     D2E.SPR
<D>ELETE        <Q>UIT
<S>WAP          <SP> TOGGLE PAGES <■>
```

```
16 x 16 SPRITE      90     D2E.SPR
→.←, A, Z  PLACE CURSOR  SPRITE#1
< RET >  TO ENTER
```

NOTE : CURSOR IS ALREADY OVER SPRITE ONE

PLACE CURSOR OVER SPRITE FOUR

```
16 x 16 SPRITE         90     D2E.SPR
→.←, A, Z  PLACE CURSOR  SPRITE#4
<RET> TO ENTER                SPRITE#1
```

```
16 x 16 SPRITE      90     D2E.SPR
<D>ELETE        <Q>UIT
<S>WAP          <SP> TOGGLE PAGES <■>
```

USE THE SPACE BAR TO TOGGLE BETWEEN THE TABLE ON SCREEN AND THE OTHER IN MEMORY

```
16 x 16 SPRITE      90     D2E.SPR
→.←, A, Z  PLACE CURSOR  SPRITE#1
< RET >  TO ENTER
```

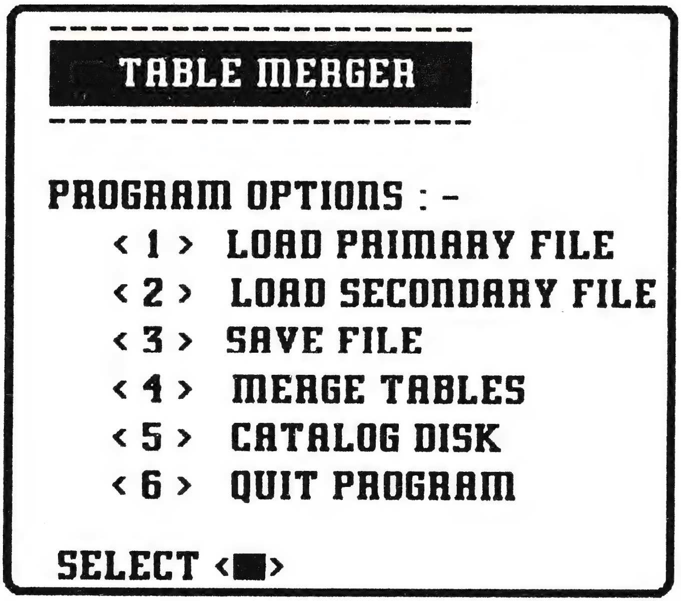
NOTE : CURSOR IS ALREADY OVER SPRITE ONE

```
16 x 16 SPRITE         90     D2E.SPR
→.←, A, Z  PLACE CURSOR  SPRITE#1
ARE YOU SURE?  ■
```

SPRITE WITHIN CURSOR IS THEN DELETED

SECTION 4.2

FLOW CHART NUMBER 2



```
----------------------------
   TABLE MERGER
----------------------------

PROGRAM OPTIONS : -
    < 1 >  LOAD PRIMARY FILE
    < 2 >  LOAD SECONDARY FILE
    < 3 >  SAVE FILE
    < 4 >  MERGE TABLES
    < 5 >  CATALOG DISK
    < 6 >  QUIT PROGRAM

SELECT <█>
```

TABLE MERGER

The Table Merger, like the two previous programmes is menu driven. As its name indicates this programme is used to merge two tables of Sprites into one. You will find this very useful when you are using Spritesoft and wanting to use Sprites from a table other than the one you have loaded. This programme allows you to select the Sprites you need from any of your tables and make up one special table from which to work.

Once again, the best way to learn is by doing. Try out both Complete and Partial mergers, (note the differences). Flow Chart number 3 on page 5.2 may help you to follow each option in this programme.

There is one point to remember about this programme if you have not already discovered it for yourself. Sprites can only be moved from the Secondary file to the Primary file.

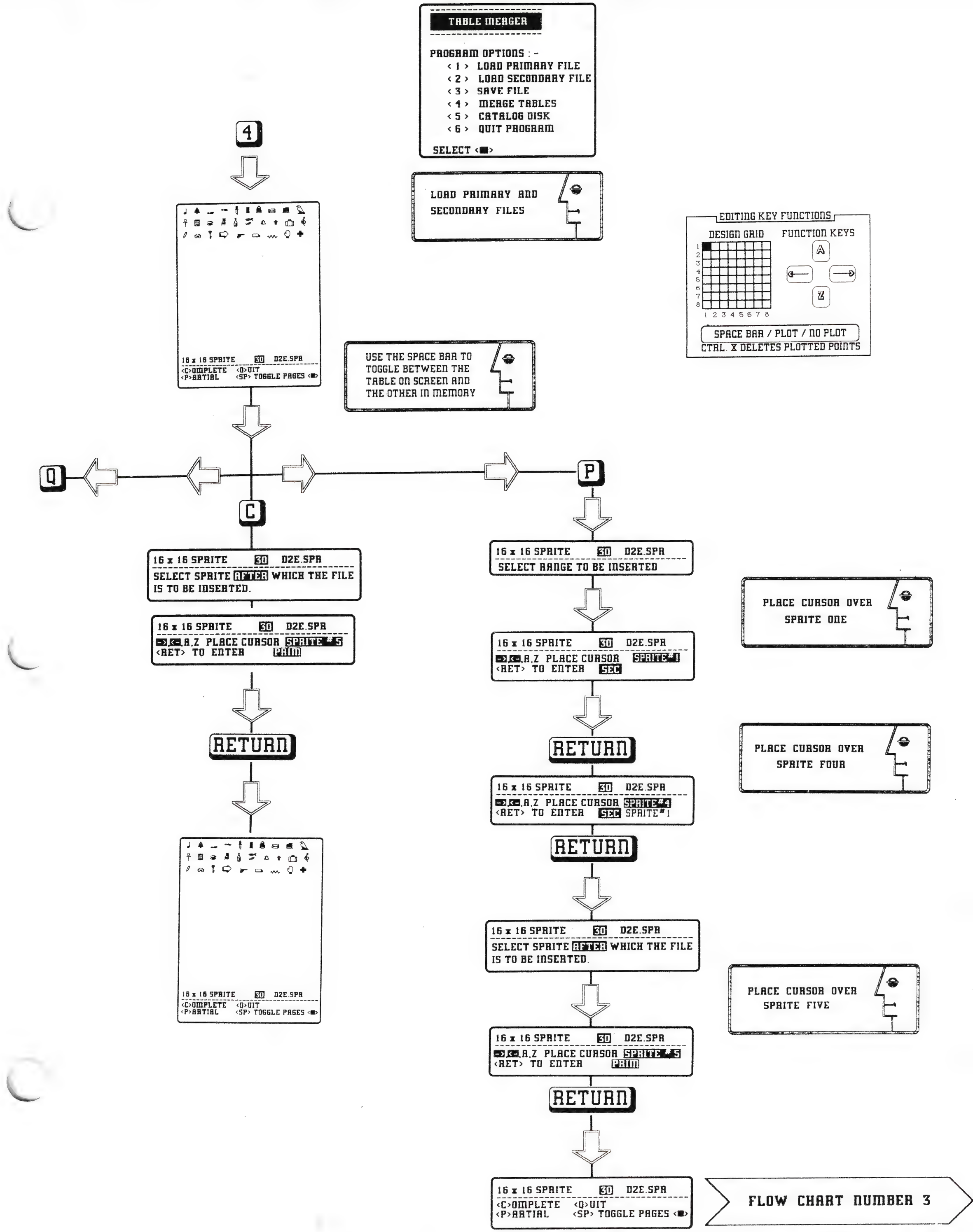


TABLE MERGER

PROGRAM OPTIONS : -
< 1 > LOAD PRIMARY FILE
< 2 > LOAD SECONDARY FILE
< 3 > SAVE FILE
< 4 > MERGE TABLES
< 5 > CATALOG DISK
< 6 > QUIT PROGRAM

SELECT <■>

LOAD PRIMARY AND SECONDARY FILES

EDITING KEY FUNCTIONS

DESIGN GRID — FUNCTION KEYS

SPACE BAR / PLOT / NO PLOT
CTRL. X DELETES PLOTTED POINTS

USE THE SPACE BAR TO TOGGLE BETWEEN THE TABLE ON SCREEN AND THE OTHER IN MEMORY

```
16 x 16 SPRITE      30    D2E.SPR
<C>OMPLETE    <Q>UIT
<P>ARTIAL     <SP> TOGGLE PAGES <■>
```

```
16 x 16 SPRITE      30    D2E.SPR
SELECT SPRITE AFTER WHICH THE FILE
IS TO BE INSERTED.
```

```
16 x 16 SPRITE      30    D2E.SPR
→,←,A,Z PLACE CURSOR  SPRITE # 5
<RET> TO ENTER         PRIM
```

RETURN

```
16 x 16 SPRITE      30    D2E.SPR
<C>OMPLETE    <Q>UIT
<P>ARTIAL     <SP> TOGGLE PAGES <■>
```

```
16 x 16 SPRITE      30    D2E.SPR
SELECT RANGE TO BE INSERTED
```

PLACE CURSOR OVER SPRITE ONE

```
16 x 16 SPRITE      30    D2E.SPR
→,←,A,Z PLACE CURSOR  SPRITE#1
<RET> TO ENTER         SEC
```

RETURN

PLACE CURSOR OVER SPRITE FOUR

```
16 x 16 SPRITE      30    D2E.SPR
→,←,A,Z PLACE CURSOR  SPRITE#4
<RET> TO ENTER    SEC  SPRITE#1
```

RETURN

```
16 x 16 SPRITE      30    D2E.SPR
SELECT SPRITE AFTER WHICH THE FILE
IS TO BE INSERTED.
```

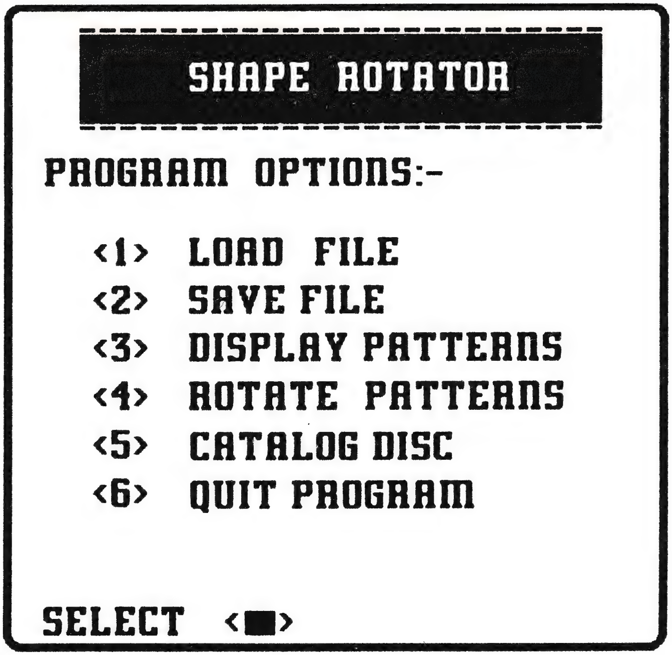
PLACE CURSOR OVER SPRITE FIVE

```
16 x 16 SPRITE      30    D2E.SPR
→,←,A,Z PLACE CURSOR  SPRITE # 5
<RET> TO ENTER         PRIM
```

RETURN

```
16 x 16 SPRITE      30    D2E.SPR
<C>OMPLETE    <Q>UIT
<P>ARTIAL     <SP> TOGGLE PAGES <■>
```

FLOW CHART NUMBER 3



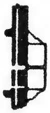
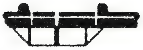
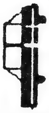
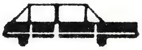
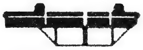
```
SHAPE ROTATOR

PROGRAM OPTIONS:-

   <1>  LOAD  FILE
   <2>  SAVE FILE
   <3>  DISPLAY PATTERNS
   <4>  ROTATE  PATTERNS
   <5>  CATALOG DISC
   <6>  QUIT PROGRAM

SELECT  <■>
```

ROTATE SHAPE

The Rotate Shape programme is similar in use to the Shape View/Sort programme. The difference between the two is that instead of sorting the patterns you use this routine to rotate them. You are also able to make mirror images of the original pattern.

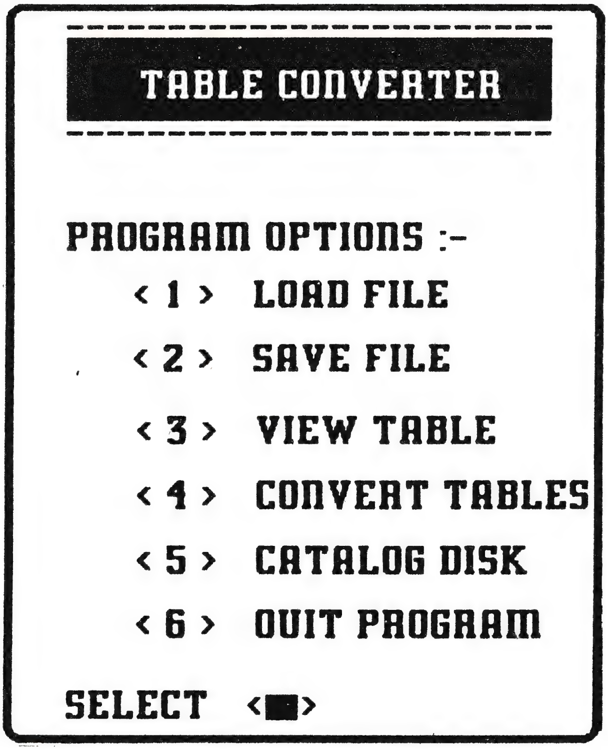
You will find that being able to rotate shapes will save you having to re-draw them. For example, you may have a car drawn which faces to the left but you want the car to drive to the right. This means that you can either back the car across the screen, re-draw the entire car or simply use the mirror image left/right routine to face the car in the desired direction.

Below is a table giving examples of each type of rotation and reflection available from this programme. Pay particular attention to the difference in results between mirror left/right and rotate 180 degrees. Also note the differences between mirror upside-down and rotate 180 degrees.

NORMAL	COMMAND	RESULT
	ROTATE <1> 90	
	ROTATE <2> 180	
	ROTATE <3> 270	
	MIRROR <L>EFT/RIGHT	
	MIRROR <U>P SIDE DOWN	

It is important to note that the Rotate Shape programme may be used with both Sprites and Shapes but you will lose the Sprites' attributes when rotating it. This means that after rotating a Sprite you will have to define its colour and position again. This does not happen when rotating Shapes as they do not have attributes.

If you cannot remember what the difference is between Sprites and Shapes or what attributes are refer to pages 2.0 and 2.1 .



```
TABLE CONVERTER

PROGRAM OPTIONS :-
    < 1 >   LOAD FILE
    < 2 >   SAVE FILE
    < 3 >   VIEW TABLE
    < 4 >   CONVERT TABLES
    < 5 >   CATALOG DISK
    < 6 >   OUIT PROGRAM

SELECT   <■>
```

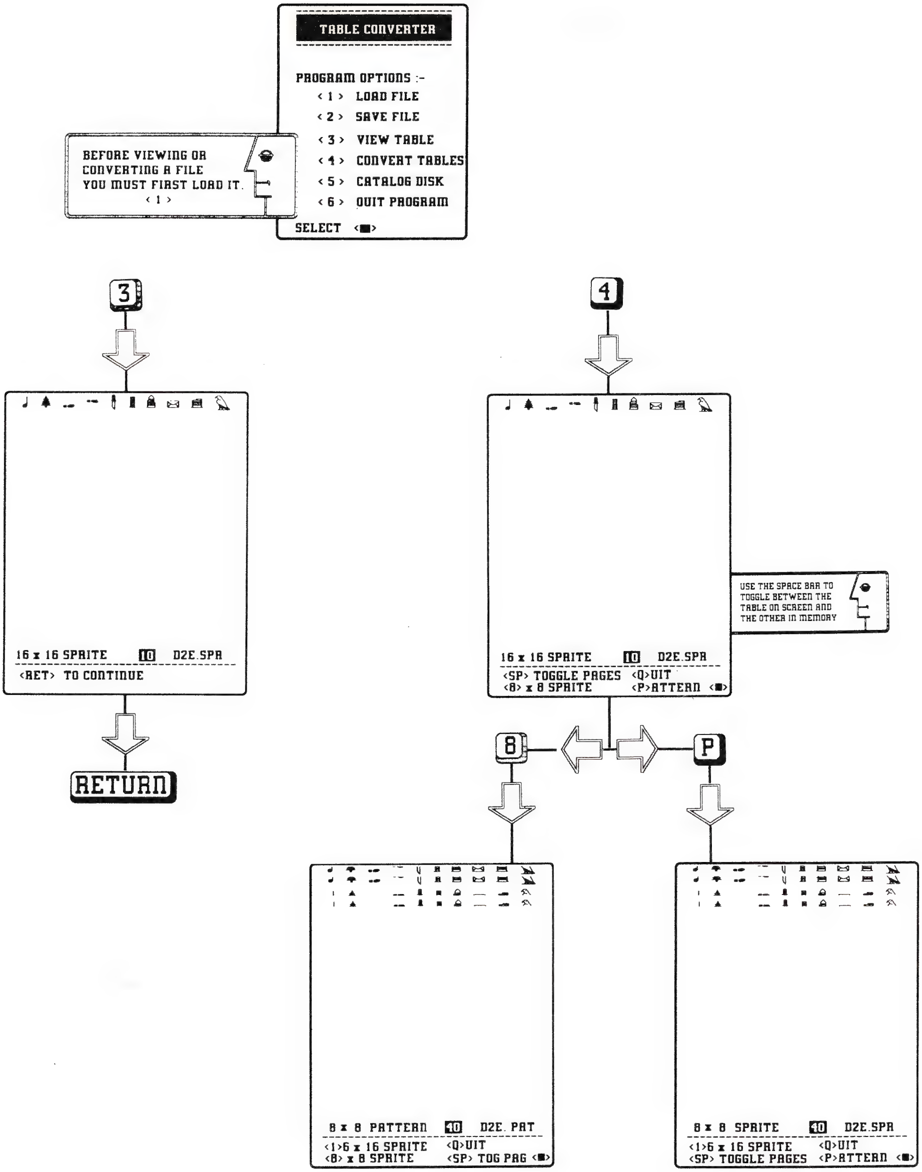
TABLE CONVERTER

Once again this programme, like the previous programmes is menu driven. The Table Converter programme is used to convert tables of Sprites from one size to another. A 16 x 16 Sprite file can be converted to an 8 x 8 file and an 8 x 8 file can be converted to a 16 x 16.

If you convert a 16 x 16 file with ten Sprites in it down to an 8 x 8 file it will then have fourty Sprites in it. Do you understand why? It is because it takes four 8 x 8 Sprites to make up one 16 x 16 Sprite. It also works back the other way; if you convert an 8 x 8 Sprite file with fourty Sprites in it up to a 16 x 16 Sprite file you will be left with a file of ten Sprites. This may be expressed mathematically:

$$10 \times 16 \times 16 - 40 \times 8 \times 8$$

Have you been asking yourself why you would want to convert a file at all. There are a number of reasons why you may want to. For example in a game you may wish to create an explosion effect. One way of achieving this would be to convert 16 x 16 Sprites in to four 8 x 8 Sprites thus exploding each Sprite in to four pieces.



```
TABLE CONVERTER

PROGRAM OPTIONS :-
   < 1 >  LOAD FILE
   < 2 >  SAVE FILE
   < 3 >  VIEW TABLE
   < 4 >  CONVERT TABLES
   < 5 >  CATALOG DISK
   < 6 >  QUIT PROGRAM
SELECT  <■>
```

BEFORE VIEWING OR CONVERTING A FILE YOU MUST FIRST LOAD IT. < 1 >

```
16 x 16 SPRITE      10   D2E.SPR
<RET>  TO CONTINUE
```

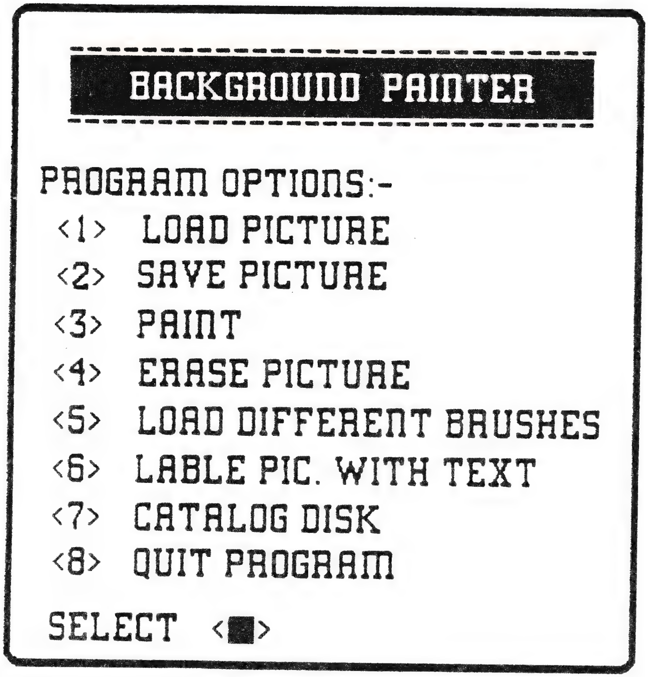
```
16 x 16 SPRITE      10   D2E.SPR
<SP> TOGGLE PAGES   <Q>UIT
<8> x 8 SPRITE      <P>ATTERN  <■>
```

USE THE SPACE BAR TO TOGGLE BETWEEN THE TABLE ON SCREEN AND THE OTHER IN MEMORY

```
8 x 8 PATTERN      40   D2E. PAT
<1>6 x 16 SPRITE   <Q>UIT
<8> x 8 SPRITE     <SP> TOG PAG <■>
```

```
8 x 8 SPRITE        40   D2E.SPR
<1>6 x 16 SPRITE    <Q>UIT
<SP> TOGGLE PAGES   <P>ATTERN  <■>
```

FLOW CHART NUMBER 4



```
BACKGROUND PAINTER

PROGRAM OPTIONS:-
 <1>   LOAD PICTURE
 <2>   SAVE PICTURE
 <3>   PAINT
 <4>   ERASE PICTURE
 <5>   LOAD DIFFERENT BRUSHES
 <6>   LABLE PIC. WITH TEXT
 <7>   CATALOG DISK
 <8>   QUIT PROGRAM

SELECT  <█>
```

PAINT BACKGROUND

The background painting programme is used to paint a picture or background scene on the Pattern plane. The diagram on page 2.2 shows the Pattern plane's position in relation to all other planes.

Once again this programme like the previous programmes is menu driven and user friendly. However the painting programme uses both the Apple's video output to display commands and options, and the Sprite's video output to display the painting surface, colour pallette and brush selections. This poses no problem if you are using either an RGB or NTSC system. These systems use the on-board soft switch to switch or toggle between screens. If you are using the PAL system without the optional soft switch you must continually change the monitor lead to the appropriate video output.

To paint a picture select option number three:

‹3› PAINT

from the main menu.

This paint routine enables you to paint on the Pattern plane with the aid of a joy stick. It also enables you to select the brush type and colour that you paint with. A joystick is used to move the cursor (paint brush) on top of one of the sixteen available colours in the colour pallette. When the cursor is on the desired colour click button zero (0) on the joy stick. This selects the colour underneath the cursor and makes a beeping sound when the selection is successful. Next select a brush from those displayed below the colours. This is done in the same manner as the colour selection.

An alternative brush set is available if you return to the main menu by selecting **CTRL-Q** , then select ‹5› and then enter the name of either of the two available brush sets: **BRUSHES1** or **BRUSHES2** and press **RETURN**

The escape key is used to toggle the Sprite's video display between the colour pallette/brush type selection and the Pattern plane painting surface. When the Pattern plane is displayed painting is achieved by holding down button zero and at the same time moving the brush. Note: both colour and brush type must be selected before attempting to toggle to the Pattern plane. Failure to select these before toggling to the Pattern plane will cause the computer to beep.

To load, save or erase any picture select options ‹1›, ‹2› or ‹4› respectively.

Note: you may notice when painting one colour on top of another that the colours bleed or spread, this is unavoidable.

If you want to label a picture or simply write on the pattern plane select:

‹6› LABEL PICTURE WITH TEXT

from the main menu. Then, when prompted by the computer press any key.

Next you will be asked to enter the file name of the font (type of letters) to be used. You have two options, enter either **ASCII.PAT** or **CHAR.PAT** The first font has both upper (ABCD) and lower (abcd) case letters. The second has large and small upper case letters but no lower case letters. Both of these files are 8 x 8 files. You must remember this as you will next be prompted to enter in the character size.

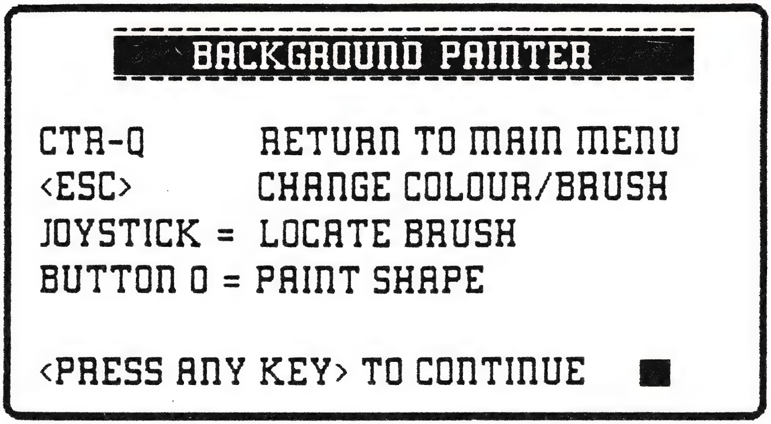
If you wish you can create your own 16 x 16 character set with the aid of the Shape Editor programme. However this programme produces Sprites not patterns and the character file must be a pattern file. Therefore you must convert your character set into a pattern file. This is achieved by selecting **‹4›** then **‹P›** from the **File Type Convert** programme and then re-saving the file to disk. Note: the characters in your character set must be in the same order as the characters in the two files provided.

After selecting the desired font and entering its size you must select a colour. The joy stick is again used to do this. Once a colour is selected the next thing to do is toggle to the Pattern plane by using the escape key and position the cursor at the point where you wish the text to start. Button zero is used to fix or lock the cursor at the desired starting point. Having done this you can type in your text. Note: it is not possible to backspace or use the carriage return in this routine. If you make a mistake you can erase it by colouring over it with black when using the paint routine.

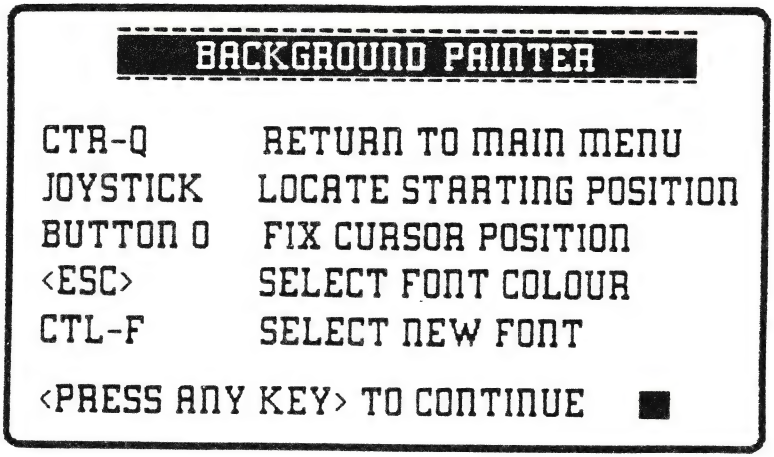
If you wish to change to another font enter **CTRL F** and the file name of the desired font and then its size. Once the file has been loaded into the computer select a colour and then press the escape key to return to the Pattern plane's painting surface.

The X and Y co-ordinates of the cursor are displayed on the Apple's output. You will find this useful when you need to locate the cursor on the same line as before.

Note: You will find that letters bleed into the coloured background unless the area behind and around the letters is a different colour.



```
BACKGROUND PAINTER

CTR-Q          RETURN TO MAIN MENU
<ESC>          CHANGE COLOUR/BRUSH
JOYSTICK =     LOCATE BRUSH
BUTTON 0 =     PAINT SHAPE

<PRESS ANY KEY> TO CONTINUE    ■
```



```
BACKGROUND PAINTER

CTR-Q          RETURN TO MAIN MENU
JOYSTICK       LOCATE STARTING POSITION
BUTTON 0       FIX CURSOR POSITION
<ESC>          SELECT FONT COLOUR
CTL-F          SELECT NEW FONT

<PRESS ANY KEY> TO CONTINUE    ■
```

VDP MODE ERROR

The computer has not been told that you are working in graphics two mode. This means that **&OG2** has not been entered.

VDP MEMORY ERROR

This occurs in graphics mode two when the memory map is inappropriate for correct pattern plane operation.

SPRITE MAGNIFICATION ERROR

This occurs if **&EL** has been entered prior to **&D,** .

Spritesoft is a language written especially for Sprites. It is used to animate, colour and control Sprites from within an Applesoft programme. We will explain this in more detail later in the manual.

To help explain Spritesoft we shall implement the use of **bold** print and square brackets [ ] . Bold print shall be used to both emphasize important points and highlight text that is written in this manual as it appears on the screen. Brackets shall be used to indicate variables. Variables are such things as numbers and words that you enter into the computer to achieve a particular result, eg. define a Sprite.

Within Spritesoft there are fifteen basic commands. Each of these commands **must be** preceded by the ampersand (**&**) character and followed by a comma (**,**) . A command will take the form (look like) :

&ML,

This particular command means **M**ove **L**eft, . Following the comma are the **expressions**. These are the terms used to tell the command what to do and what to do it to. For example, two expressions may be used with the move left command to : 1. Identify a particular sprite plane

and

2. Tell it how far to move it to the left.

That is :

&ML,[Plane],[Distance]

This may sound confusing but is easy to follow when we apply this to a practical situation. For exampe: **&ML,4,10**

means

&Move to the **L**eft, Plane Number **4**, A distance of **10** spaces (Pixels)

Not too difficult to follow is it?

The following pages explain each of the Spritesoft commands. You may wish to try a command out after having read about it. Why not? The best way to learn about computers is by using them, so give it a try. The worst that can happen is that nothing will appear on the screen. However before using Spritesoft you must load it into the computer. This is achieved by selecting:

<I> INSTALL SPRITESOFT

from the main menu. If you are not sure which is the main menu refer to the diagram on page 3.0 .

After having done this, and assuming you have installed your Sprite Graphics Card correctly, you can proceed to the next page, COMMANDS

&C,[COLOR] Change backdrop colour. See Backdrop plane, page 2.1 . Must have a value from 0 to 15. See colour chart page 9.7.

This command followed by colour number 6 eg. **&C,6** will, if your system has been installed correctly, cause the entire screen to turn dark red.

&L, [File name] Loads the desired file .

This carries out the loading of Sprite and Pattern files from the disk to the memory. **NOTE** When this instruction is used directly from the keyboard a "Syntax Error" will appear on the screen after the command is executed. This is normal and does not happen in a programme as long as the file name ends with SPR or PAT . A file containing a table of Sprites must be loaded before any of the following commands are able to be used.

&T,[Sprite size] Defines the Type (Size) of Sprite that you are going to use.

Spritesoft must be told the size of the Sprites that are to be used. If you are loading 8x8 Sprites then enter **&T,8** and if you are loading 16x16 then enter **&T,16** .

&S,[Sprite], [Plane], [X], [Y], [Color] Sets the defined Sprite onto the defined plane at the position defined by the X and Y co-ordinates in the defined color.

[Sprite] represents the number of the Sprite in the loaded Sprite table. This could be any number from 1 to 128 if you have a completely full file of 8 x 8 Sprites.
[Plane] represents the plane number on which you wish to place the Sprite. Only one Sprite per plane is permissible (allowed). Plane numbers start at 0 and go to 31, giving 32 planes in all.
[X] ,**[Y]** represents the co-ordinates on the screen at which the top left hand corner of the Sprite is set. The X co-ordinate may be any number from 0 to 255; the Y co-ordinate, any number from 0 to 191.
[C] represents the color of the Sprite and may be any number from 0 to 15. Refer to the color guide on page 9.7

&S,[Sprite], [Plane], [X], [Y], [Color]

When using this command it is not essential to define the last three expressions, X, Y and Colour. If these are not specified their default values will be used. Defaults are those values or Attributes defined when the Sprites were originally created with the Sprite Editor routine.

What does it all mean? Just this : any of the following three commands will work correctly.

&S, [Sprite], [X], [Y]
&S, [Sprite], [Plane], [Color]
&S, [Sprite], [Plane]

These variations of a command are called **Permissible Forms**. They are very useful when setting a Sprite which has suitable default values .

&P, [Plane] Locates the X and Y position of the top left hand corner of the Sprite on the specified plane.

You will find this command very useful when you wish to locate the position of one Sprite in order to place another next to it.

&EL Enlarges all Sprites.

This command magnifies all Sprites by two. This means that an 8 x 8 Sprite becomes 16 x 16 and a 16 x 16 Sprite becomes 32 x 32.

&ES Shrinks all Sprites.

This shrinks all **previously enlarged** Sprites back down to their original size.

&M, [Plane], [X], [Y] **Moves the Sprite on the defined plane to the defined X, Y co-ordinates.**

This command may be used to make a Sprite appear to jump from one position to another. All movement commands can move up to four planes at one time. This means that this command has four **Permissible Forms**. These are :

&M, [Plane], [X], [Y]
&M, [Plane], [Plane], [X], [Y]
&M, [Plane], [Plane], [Plane], [X], [Y]
&M,[Plane], [Plane], [Plane], [Plane], [X], [Y]

For example : **&M,1,2,128,96** moves the Sprites on plane numbers 1 and 2 to the x , y co-ordinates 128, 96, which is the centre of the screen.

&M Direction, [Plane], [Distance] **Moves the Sprite(s) on the defined plane(s) Up, Down, Left or Right over a defined Distance.**

The four directions of movement make up four permissible forms of the movement command :

&MU, [Plane], [Distance]
&MD, [Plane], [Distance]
&ML, [Plane], [Distance]
&MR, [Plane], [Distance]

For example : **&ML,0,1,2,3,2** will move Sprites on planes 3, 2, 1 and 0 (in that order) to the left by 2 pixels. Negative distances may be specified. For example:

&MR,3,-10

Moves the Sprite on plane 3 a distance of 10 negative pixels to the right. This means that the Sprite moves a distance of 10 positive pixels to the left. Therefore :

&MR,3,-10 is the same as **&ML,3,10**

Don't forget that each of these commands can move as many as four planes, ie. four Sprites with the one command. The distance moved is up to you. Any distance may be specified from one pixel to a complete lap of the screen. Try experimenting with different distances and see what happens.

This command (as with all commands) may be placed within an Applesoft routine such as a FOR NEXT loop and used as an animation routine (see page 3.4). Later, in the programming section, this will be dealt with in greater detail.

&MA, [Plane], [Angle], [Distance] Moves the Sprite on the specified plane at the specified angle over the specified distance.

The Angle expression is defined in terms of minutes. That is any number from **1 to 60**. This works in a similar way to the minute hand on a watch face. For example :

&MA,3, 0,15,20

Moves Sprites on planes 3 and 0 off at an angle of 15 minutes. The same angle as the minute hand at fifteen minutes past the hour. If you have not already noticed it for yourself, **&MA,15** is also the same as **&MR,** . See if you can find other commands and expressions which duplicate (copy) each other.

For those who are mathematically minded you may wish to note that one minute is equal to six degrees. This fact may be useful later when writing a program to move a Sprite over a particular type of shape. A pentagon for example is made up of five equal angles of seventy two degrees or twelve minutes.

&MV,[Plane],[X Distance],[Y Distance] Move Vector shifts Sprite(s) on the specified plane(s) to the point along the vector determined by the X and Y distances.

Move Vector is similar in effect to &MA except that instead of specifying an angle, X and Y distances are given. **Note**: negative distances may be specified. For example:

&MV,1,2,10,-20

Moves Sprites on planes 1 and 2 to the right (a positive direction) by 10 pixels and down (a negative direction) by 20 pixels.

&BL,[Plane] The Sprite on the defined plane will Bleed Left off the edge of the screen.

Look at the demonstration programme of the car scene and pay particular attention to the Sprites entering and disappearing off the screen. You should have noticed that when Sprites enter or leave the edge of the screen, they either do so a bit at a time (they **Bleed** on or off) or all at once (they jump on or off). If you have written a movement programme and you want a Sprite to bleed off the screen instead of jumping you must use the Bleed command. Up to four Sprites may be specified with the one command. If a Sprite bleeds off the screen it will jump on when it next appears and vice-versa.

Note: Due to the design of the graphics processor Sprites will jump to the left or right by 32 pixels when commanded to bleed left or right respectively. This should be kept in mind when originally setting these Sprites.

&BR,[Plane] The Sprite(s) on the defined plane(s) will bleed right, off the screen.

This command is the same as &BL except that it is used to bleed Sprites off to the right.

&H,[Plane],[Plane],[Variable] Sets either a o or 1 when defined planes **Hit** .

This command is used as a hit or collision detector. When the two Sprites on the defined planes overlap by one pixel the variable indicates a **1** . At all other times the variable indicates a **0** .
Note: A hit is detected when any coloured or transparent part of an 8x8 or 16x16 or 32x32 Sprite matrix overlaps with another.

&CS,[Plane],[Color] Colours the defined **Sprite(s)** with the desired colour(s).

This command is used to change the colour of Sprites once they are set on the screen. Up to four Sprites may be changed with the one command. This means that this command has up to four permissible forms:

&CS,[Plane,[Colour]
&CS,[Plane],[Plane],[Colour],[Colour]
Etc.
Etc.

&W **W**ipes the Sprite card's memory clean.

This is used to erase the VDP (Sprite card's) memory clearing everything from the Sprite's video display **except** the Backdrop plane.

&I Initialises the Sprite Graphics Card.

This is executed automatically when Spritesoft is installed. If you wish to clear the **entire** screen and the computers memory the **&I** command may be used to do this.

COLOUR TABLE

0	-	TRANSPARENT
1	-	BLACK
2	-	MEDIUM GREEN
3	-	DARK GREEN
4	-	DARK BLUE
5	-	LIGHT BLUE
6	-	DARK RED
7	-	CYAN
8	-	MEDIUM RED
9	-	LIGHT RED
10	-	DARK YELLOW
11	-	LIGHT YELLOW
12	-	DARK GREEN
13	-	MAGENTA
14	-	GREY
15	-	WHITE

You will find this section easy to understand if you have already read section 8, "Paint Background" and section 9 "Spritesoft".

Paintsoft is an extension of Spritesoft and utilises the ampersand character (**&**) in the same manner as Spritesoft. Paintsoft may be used to: transfer Sprites to the Pattern plane, save and load the pictures you have painted and organise both the computer's memory and the Sprite's memory. To help explain Paintsoft we shall implement the use of **bold** print and square brackets [ ] as we have done in the previous Spritesoft section.

This section explains each of the Paintsoft commands. You may wish to try a command after having read about it. Go right ahead, it is the best way to learn. But be warned, you may lose the picture you are working with if you save it incorrectly. Practise with a picture that you will not be upset if you lose.

You must load Paintsoft into the computer's memory to enable you to use it. This is achieved by selecting:

‹A› INSTALL PAINTSOFT

from the main menu. After having completed the installation you can proceed to the next page and try out (execute) the commands.

NOTE: The Paintsoft programme is partially written in Spritesoft and therefore both Paintsoft and Spritesoft are loaded when you install Paintsoft. This means that all Spritesoft commands are available for use within Paintsoft.

&OG2 Tells the computer that you are going to **O**perate in a mode called **G**raphics **2**

You **must** tell the computer that you are going to work in the Graphics 2 mode if you want to work on the pattern plane.

&F,"[Filename]" Saves the picture to the disk under the desired filename.

The procedure of saving files in Paintsoft differs from Spritesoft in an important way. When you save a Spritesoft file, ie. a file ending with .SPR you only work with that single file. When you save a Paintsoft file/picture the computer automatically saves three files. These three files contain tables which define:

1. Its colour- Colour Table, this file when saved to disk will automatically end with **.CT**
2. Its patterns- Pattern Table, this file when saved to disk will automatically end with **.PT**
3. Its name- Name Table, this file is the same for all picture files and there is no need for you to save it.

These commands generate (make) an error message which appears on the screen when entered directly from the keyboard, as opposed to being executed from within a programme. Do not be concerned, the reason this occurs is that Applesoft does not understand these commands. They will however be executed correctly despite the error message.

&L,"[Filename]" Loads the desired file into the computers' memory.

This loads into the **computer's memory** only **one** of the three tables needed for each picture. This command must be used twice. Once to load the Colour Table and again to load the Pattern Table. (A different command is used to load the name table and will be described later.)
Below is an example of loading a picture named "Car Scene" :

&L,"Car Scene .CT"
&L,"Car Scene .PT"

The commands on the previous page load the colour table and the pattern table into the computers memory. Note: both tables still need to loaded in the Sprite's memory.

&RC
&RP

Loads Colour and Pattern files respectively into the Sprite's VDP memory.

These two commands are used to transfer the tables you have loaded into the computer's memory into the Sprite's VDP memory making them available for display. The most efficient way of using these commands is to place them after a colon following the command which loads the tables into the computers memory, eg.

```
&L,"Car Scene .CT":&RC
&L,"Car Scene .PT":&RP
```

This will load both Colour and Pattern tables into the computer's memory and then into the VDP memory.

&RN Loads the Name table directly into the VDP memory.

As the Name table is the same for all Pattern files it is not necessary to define a filename for the name table. This also makes it possible to load this table directly into the Sprite's VDP memory. This means that you do not have to load it into the computer's memory first.

Below is an example of a short programme which will load all three files into the Sprite's VDP memory via the computer's memory and display the pattern on the screen:

```
10 &L,"Car Scene .CT":&RC
20 &L,"Car Scene .PT":&RP
30 &RN
RUN
```

&D,[Plane]

Draws the Sprite on the defined plane onto the Pattern plane.

You will find this very useful when you are attempting to reduce the number of Sprites on the screen. For example once a Sprite is stationary and is not going to be moved you can transfer it to the pattern plane.

Note: 1. Before using this command you must have already entered the &RN command and all the Sprites on the screen must be in the times one (x1) mode. An error message may occur when using this command. This is because Applesoft does not understand what it means. Once again, do not be alarmed by the error message.

2. The Sprite to be transferred must be within a visible area of the screen.

3. This command works with 8 x 8 and 16 x 16 Sprites in times one magnification only.

&W Wipes or erases everything from the VDP memory.

&WP Wipes the Pattern plane.

&WS Wipes the Sprites from memory.

&FC Reads Colour table From VDP to the computer memory.

This command is for advanced programmers and is used to read the Sprites VDP Colour table and load this information to address $2800 in the computer's RAM. The length is $1800. This command is useful when you wish to save a Colour table to disk under a particular filename.

&FP Reads Pattern table From VDP to the computer memory.

This command is also for advanced programmers and is used to read the Sprite's VDP Pattern table and load this information to address $2800 in the computer's RAM. The length is $1800. This command is useful when you wish to save a Pattern table to disk under a particular filename

&I This is used to initialise the Video Display Processor (VDP). In its basic form as shown it uses the default values loaded with Spritesoft ($35A-$361). The current register states are stored at $374 - $37B. These VDP registers are write only.

&I,[Register 1], to [Register 7]

This form may be used to set the VDP registers to any legal value. All eight registers **must** be specified as given in the Texas Instruments 9900 Data Manual.

Default Values:

Register 0	$00	0
Register 1	$00	192
Register 2	$06	6
Register 3	$00	0
Register 4	$04	4
Register 5	$36	54
Register 6	$07	7
Register 7	$21	33

These values give Graphic 1 Mode with black background, 8 x 8 Sprites with x1 magnification.

&IA,[Register No.]

Data used for setting individual registers to the data specified. A typical form is:

```
&IA,1,192
```

Sets register 1 to 192 as per default table.

&M,

As you have seen in the car demonstration scene, Sprites appear to jump off the left hand edge of the screen. This can be overcome by having the MSB of byte 4 of the Sprite's attributes set while it is in the left half of the screen and reset in the right half. Changing line 170 from "D2A.SPR" to "D2B.SPR" will run the programme with the bit set. The Sprites will then jump at the right edge of the screen.

Further changes will have to be made in the machine code routines that are written by lines 120 to 150. These test in which half of the screen the Sprite is. Then sets and resets the bit accordingly. Refer also to &BL or &BR instructions.

Machine Code Routines. All routines are relocatable in memory.

"**SPRITE ROUTONES.REL**"
Contains all the routines at normal base address of $7000.

"**TABLOAD.REL**"
Used to load all data to the VDP Ram, $1A.$1f are used as pointers. [A47000, L$24].

"**CLRVDPMEN.REL**"
Clears the VDP Ram. [A$7024, L$11]

"**LVDPREGS.REL**"
Loads the VDP register with the contents of the 6502's A and Y registers. [A$703C, L47]

"**INITVDPREG>REL**"
Initiates the VDP registers from a table of 8 bytes whose address is in pointers $1A-$1B. [A$7043, L413]

"**CHGREG.REL**"
Loads a VDP register from pointers $06 and $07. [A$7056, L$OB]

"**KBDREAD>REL**"
Similar to the GET instruction but stores the character at $19. [A$7061, L$12]

"**SPHRCON.REL**"
Used by the shape viewer to convert the Shape Table to a High Res picture. It uses $1A.1D as pointers. Normal address $313 8x8 [Left in 16x16] and $343 right in 16x16. $373-39C Hires address tables. Loads to A$7072,L$B. Result return in Register A.

"**VDPRAMREAD.REL**"
Used to read a single memory location in the VDP Ram. The Address is specified in locations $01-$02 and the result is returned in the accumulator.

Reading the VDP STATUS REGISTER is done with "AD D1 CO LDA $COD1" if the Sprite card is located in slot 5.

"SPRITESOFT"

Located at addresses $8DB0-$95FF. It uses the following Ram locations :

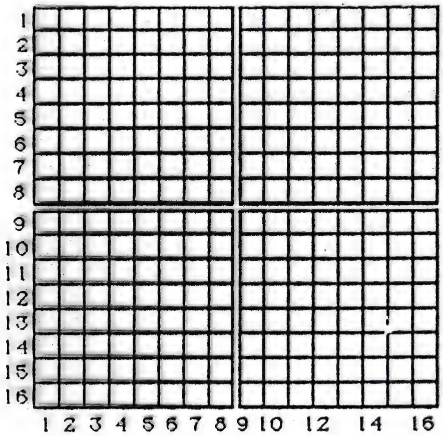
$0-$7 for calculation of offsets.

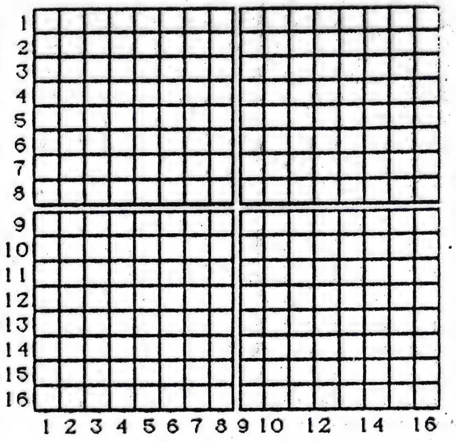
$35A-$38F for VDP register, default values, memory map and input buffer.

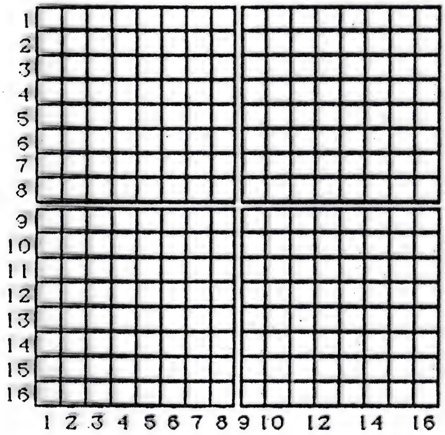
$310-$327 to assemble the output for the Load (&L) command.

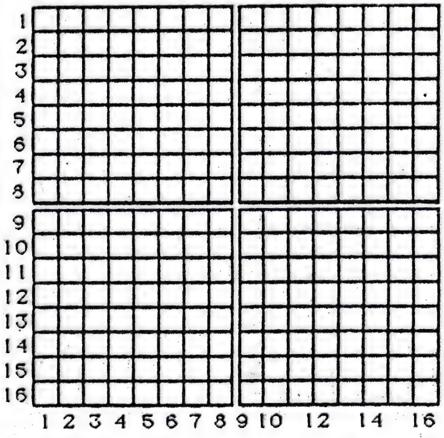
$1A-$1F as the Sprite routines.

$F9-$FF temporary data storage.


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1 2 3 4 5 6 7 8 9 10 12 14 16


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1 2 3 4 5 6 7 8 9 10 12 14 16


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1 2 3 4 5 6 7 8 9 10 12 14 16


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1 2 3 4 5 6 7 8 9 10 12 14 16

SPECIFICATIONS

Processor	TMS 99-- range
RAM	16 K DRAM Speed 250Ns.
Input RGB	Pin 1 = Red Pin 2 = Green Pin 3 = Blue Pin 4 = Ground Pin 5 = Not connected Pin 6 = Sync.
Input NTSC	Composite video NTSC
Output PAL	Composite video PAL 626 lines non interlaced 1 Volt peak to peak Frame rate 50Hz. Negative going sync.
Output RGB	As per input connections 626 lines non interlaced 1 Volt peak to peak Frame rate 50 Hz. Negative going sync.
Output NTSC	Composite video NTSC. 524 lines non interlaced 1 Volt peak to peak Frame rate 60 Hz. Negative going sync.

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE.

THE MICRO DOCTOR
1/1 Rosney street
Hawthorn, 3122
Phone (03) 819 1397 AH